

Automatic Discoveries of Physical and Semantic Concepts via Association Priors of Neuron Groups

Shuai Li

Chinese University of Hong Kong
sali@ee.cuhk.edu.hk

Kui Jia

South China University of Technology
kuijia@scut.edu.cn

Xiaogang Wang

Chinese University of Hong Kong
xgwang@ee.cuhk.edu.hk

Abstract

The recent successful deep neural networks are largely trained in a supervised manner. It *associates* complex patterns of input samples with neurons in the last layer, which form representations of *concepts*. In spite of their successes, the properties of complex patterns associated a learned concept remain elusive. In this work, by analyzing how neurons are associated with concepts in supervised networks, we hypothesize that with proper priors to regulate learning, neural networks can automatically associate neurons in the intermediate layers with concepts that are aligned with real world concepts, when trained only with labels that associate concepts with top level neurons, which is a plausible way for unsupervised learning. We develop a prior to verify the hypothesis and experimentally find the proposed prior help neural networks automatically learn both basic physical concepts at the lower layers, e.g., rotation of filters, and highly semantic concepts at the higher layers, e.g., fine-grained categories of an entry-level category.

1 Introduction

Deep neural networks [1] have recently brought remarkable breakthroughs in high dimensional perception problems such as image classification [2] and object detection [3]. Most of these successes are achieved in the context of supervised learning where tremendous amounts of labeled training samples are provided. Based on training objectives that directly *associate* training samples with their semantic labels, supervised learning learns neural networks that associate complex patterns of input samples to hidden neurons in the last layer, which form representations of *concepts*. These hidden neurons are subsequently fed to a classifier to estimate the probability of semantic labels. In spite of their successes, the properties of complex patterns associated a learned concept remain elusive.

In this work, by analyzing how neurons are associated with concepts in supervised networks, we hypothesize if with proper priors to regulate the learning of neural networks, networks can associate filters with concepts without supervised labels, which is a plausible approach to unsupervised learning. To reach this hypothesis, we analyze how neural networks associate neurons with concepts, as illustrated in Figure 1. We define Concept formally in Section 2, which allows us to analyze how a neuron is built as a representation of a concept in neural networks in Section 3. The analysis implies that supervised labels can be understood as a prior that associates neurons with concepts sample by sample, and leads to our hypothesis. To initially investigate the hypothesis, we observe that a rich hierarchical structure exists in concepts — concepts are built on top of variations of concepts, which in turn is a concept themselves, e.g. the concept of aquatic animals is built on top of fishes, seals etc, and fishes is a variation of aquatic animals. We further hypothesize that with proper priors to regulate learning, neural networks can automatically associate neurons in the intermediate layers

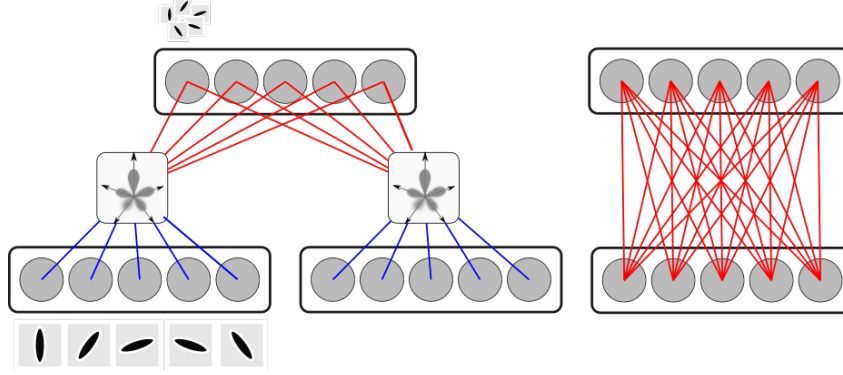


Figure 1: Illustration of Concept; of Neuron as representation of Concept; and of the proposed prior to regulate learning of Neural Network and its comparison with traditional neural networks. Concept is the cluster of images that contains variations of a prototype, e.g. the concept of an edge that consists of various orientations of an edge, shown at the lower left corner, and technically defined as the orbit generated by symmetry groups. Now we take orbit, concept and cluster as synonyms. Neurons in this layer are variations of concepts of neurons in the next layer. A neuron in the next layer is built through convolution i.e. the red lines, and serves as the representation of a concept that is associated with/invariant to variations of the concept (thanks to cross channel summation in convolution), e.g. as the concept shown above the top left neuron, a neuron in the next layer may be the concept of an edge, and will be active regardless of which orientation of edges is present. Illustrated on the left, we regulate the learning of neural networks with a prior that divides neurons into groups, and endorse the statistic prior on them that only one neuron/variation will be active, denoted as the probability density cloud in the square frame. Each arrow in the frame denotes a variation of a concept, and the probability density denotes how probably the current input sample contains that variation. Compared with traditional neural network, shown on the right, interaction among neurons in the same layer is present, denoted as blue lines. In modeling their interaction, we hope we can group variations of the same concept into the same group, e.g. all neurons in the group represent edges with various orientations. These variations are themselves more fine-grained concepts, which we consider may be aligned with real world concepts even if we do not train networks with labels of these concepts.

with variations of a concept that are aligned with real world concepts, when trained only with labels that associate concepts with top level neurons. To verify, we develop a prior to regulate the learning of networks in Section 4. It divides neurons into groups, and aim to group variations of the same concept in the same group via promoting competition among them. Then by only providing labels at the top granularity of a concept, we test whether concepts of lower granularity can also be learned without direct labels. Related works are discussed in Section 5.

In Section 6, we report experimental results. We train supervised neural networks with the proposed prior, and investigate learned filters and neurons to see whether they can be associated with known concepts. By applying the prior to a neural network, we find through filter visualization filters in the same group of the first layer tend to be physical variations of each other, the relationship among which probably forms a symmetry group. In a problem where classes of training samples have fine-grained class structures (i.e. a class has its underlying sub-classes), with proposed prior, we find neural networks are able to associate a filter with a sub-class variation within a super-class automatically (in this problem case, we also call the classes with training labels as common or super classes). More specifically, the prior is applied on the penultimate layer. The last layer is a degenerated linear filter (max pooling) layer that computes representations of super-class concepts. Training labels associate neurons there with concepts of super-classes. After training, we find neurons in the penultimate layer, which are trained with no labels, discover clusters of samples that are particularly relevant to sub-class variations in the super-class. When using discovered neurons to classify test samples, we obtain classification performance that is far beyond the accuracy of chance and approaches that of fully supervised learning. In contrast, no clusters relevant to the sub-classes emerge from internal neurons of neural networks without using our proposed association prior.

2 Concept

In this section, we define **Concept** using symmetry group theory.

Concept as variations of a prototype. First, we give an informal definition that tries to capture the intuition behind the formal definition given later. We define a cluster of images that consist of a prototype t and its variations $t', \forall t' \in T$ as the concept related to that prototype, where T is the set of all variations. To ease discussion following, we call t' a variation of a concept. It models the behavior that humans distinguish or learn a novel concept by remembering typical properties of a prototype of such concept, and are able to generalize to its variations, e.g. an apple and its various viewpoints.

Symmetry group. To give a formal but limited definition to t, t' , we introduce Symmetry Group Theory. A symmetry of an object is a transformation that leaves certain properties of that object intact [4]. If we have two symmetry transformation g and h and we compose them, the result gh is also a symmetry transformation. A symmetry group is a set with such a composition operator on its elements. The composition needs to satisfy closure and associativity. Identity element and inverse of any member also need to exist.

Concept as orbits of symmetry group. With the machinery of Symmetry Group, we define a concept as the orbit of a prototype generated by certain symmetry groups. For a symmetry group G , the orbit is a cluster of images consists of the prototype t , and its variations $gt, \forall g \in G$, where g is the operation that varies/transform t in group G . The definition characterizes the relationship between t and t' . For strict subsets of variations that are induced by known symmetry groups, e.g. rotation, of a prototype t , such definition is strictly accurate. In this case, we have a clear definition of g, G, t , e.g. a rotation

translation group can be represented through matrix, $g(r, u, v) = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix}$,

$G = \{g(r, u, v) | r, u, v \in Z^3\}$, and t is a function $t : Z^3 \rightarrow R^{|Z^3|}$ defined on Z^3 . Z^3 is a tuple composed of three integers. From then on, we will treat orbit and concept as synonyms.

Limitation of symmetry groups definition. However, it is unlikely that more complex transformation has an explicit parametric form as that of a small affine group, e.g. SO rotation group given before, considering that they need more than just physical rules. As an example, the face viewpoint transformation actually needs further information, e.g. the other side of the face, which has to be learned. Probably new mathematical objects need to be created to characterize such behaviors, starting from symmetry group theory. Thus, the above formal definition is limited, but it serves as concept reuse for the coming discussion.

3 Neural network associates neurons with concepts

In this section, based on the concept defined, we explain the hypothesis how neural networks associate neurons with concepts.

3.1 Neural network

Formally, a convolutional neural network is a composition of layers of computational units which defines an operator

$$T : x(u) \rightarrow x_L(u, c)$$

where $x(u)$ is the input signal, such as an image $I \in L^2(\mathbb{R}^2)$, $x_L(u, c)$ is the last layer that would be fed to a classifier; u is the translation index, and c is the channel index. $x_L(u, c)$ is computed from $x(u)$ through composing representations layer by layer. Representations $x_l(u, c)$ at layer l is obtained from representations $x_{l-1}(u, c)$ at layer $l-1$ by applying a linear operator and an activation function:

$$x_l = \eta(W_l x_{l-1})$$

3.2 Neurons are representations of concepts

Theoretical and experimental works suggests neural networks encode complex transformation. [5] implies that instead of learning transformations g for complex variations, neural networks directly learn transformed results — orbit elements gt . The orbits are generated from discrete samplings of continuous lie groups at intervals such that the values of continuous transformation can be interpolated in between — lie groups are continuous symmetry groups whose elements form a smooth differentiable manifold — though the properties of those groups remain largely unknown. Experimentally, the community has observed and hypothesized that the feature space of neural networks lies in certain manifolds/lie groups [6] [7]. As examples, point can traverse in the feature space to move from an image of a person from a young age to a senior age [7], which demonstrates filters somehow encode such age transformation.

Inspired by above works, we hypothesize a neuron in x_l serves as a representation of a concept, where in a neural network, a cluster of samples of various variations of a concept are mapped to when doing forward propagation. The representation is built recursively as described in the following.

The cross-channel summation done when convoluting x_{l-1} with W_l is possibly a general way to selectively filter elements of orbits [5], or in another word, reuse concepts, and build new representations of concepts from them. We illustrate the effect of convolution with a degenerated linear filter — max pooling. A neuron $x_{l+1,G}$ after convoluting/linear filtering are

$$x_{l+1,G} = \max_{\forall g_i \in G} \{ \langle x_{l-1}, g_i t \rangle \}$$

where gt denotes filters in layer l , linear filtering is written as inner product, the filter in layer $l + 1$ that corresponds to the neuron $x_{l+1,G}$, whose functional form is max operator, has coefficients 1 in the dimension where x_l has the large activation value, and 0 otherwise. $\langle x_{l-1}, g_i t \rangle$ checks whether the input has orbit element $g_i t$, and the max operator computes a representation of the orbit/concept that is invariant to G .

x_{l-1}, x_l are also built similarly and is invariant to smaller symmetry groups. With a compositional structure, distributed regions in the input space can be routed to the same neuron in a feature map $x_l(u, c)$. When η is a piecewise linear activation function, [8] theoretically analyzes the expressive power of a hidden neuron in feedforward neural networks. Consequently, each neuron of x_l is capable of being invariant to a large G built hierarchically, and associates with a cluster of samples of all kinds of variations of a concept.

4 The proposed prior

According to the hypothesis, in supervised neural networks, each neuron in the last layer of the network computes a representation of a concept, e.g. Cat. Supervised labels can be understood as a prior that functions sample by sample to associate a particular neuron with a cluster of variations of cat images. An important question to ask is, is it possible to deduce more general principles underlying the sample-wise priors, and associate neurons with concepts in an unsupervised manner?

To initially explore the question, we observe that a rich hierarchical structure exists in concepts — concepts are built on top of variations of concepts, which in turn is a concept themselves, e.g. the concept of aquatic animals is built on top of fishes, seals etc, and fishers is a concept itself. We hypothesize that with proper priors to regulate learning, neural networks can automatically associate filters in the intermediate layers with variations of a concept that are aligned with known concepts, when trained only with labels that associate concepts with top level neurons.

We develop a prior to fulfil such tasks in this section. The intuition of the prior is that only when a concept is novel enough, a network bothers to generate another neuron to represent that concept. More formally, since orbit elements are generated by discrete sampling of a lie group on the hypothesis to guarantee continuous interpolation, only when two orbit elements are different with each other considerably, it needs another sampling to differentiate one from another. Given the radical variations between neurons, correspondence between neural-network-learned concepts and concepts in the real world can be found, which we experimentally verify later.

4.1 Competitive Group-restricted Boltzmann Machine

To characterize the intuition, we divide neurons into groups, and aim to group variations of the same concept in the same group via promoting competition among orbit elements that are generated by the same lie group.

It works by encouraging concepts to associate with neurons that have larger activation compared with other neurons, and be picky with orbit elements. The rationale is during training, all neurons are different slightly at the beginning by random initialization. The output of linear filtering is the auto-correlation score between an input patch (an image patch or feature map patch of the previous layer) and the filter. The larger the output, the more characteristic that neuron is for the concept of that sample, so that neuron should more associate with that sample. To put it in another way, for the concept this particular sample belongs to, it should be represented with such neuron. On the other hand, one widely acknowledged observation that one function of hidden neurons is to detect salient patterns [9], and passes relevant signals to the next layer. Thus, one potential relationship between $gt, \forall g \in G$ is to characterize this functionality by giving out the most representative gt in the orbit, consequently reducing neuron activation variability, and sparsifying activations, which may lead to better linear separability, better disentanglement of factors of variations and more representative neurons [10][5]. For example, the neurons that are associated with different viewpoints of a face does not fire at the same time (as a side note, two viewpoints of the same face in different locations of an image is another matter which is modeled by translation symmetry). After training, the competition may lead to filters that are considerably different orbit elements, and for each forward propagation, only the most salient element is passed on. In addition, competition is a canonical computation [11] in the brain.

Let the joint probability of a set of hidden units \mathbf{h} and a set of visible units \mathbf{v} of a Boltzmann machine be

$$p(\mathbf{h}, \mathbf{v}) = \frac{1}{Z} e^{\mathbf{b}^T \mathbf{h} + \mathbf{c}^T \mathbf{v} + \mathbf{v}^T \mathbf{W}^v \mathbf{h} + \mathbf{h}^T \mathbf{W}^h \mathbf{h}},$$

where \mathbf{h} and \mathbf{v} are binary vectors, and $\mathbf{W}^v, \mathbf{W}^h$ are matrices modeling the interactions between visible units and hidden units, hidden units and hidden units respectively. For simplicity of notation, a feedforward network notation is used, but indices of hidden neurons actually represent channel indices.

The competition statistic prior is endorsed by setting a fixed between-hidden-neuron weights. We divide hidden units into groups and set

$$W_{ij}^h = \begin{cases} -\infty & \text{hidden units } i, j \text{ are in the same group} \\ 0 & \text{hidden units } i, j \text{ are in different groups.} \end{cases}$$

It indicates that two hidden units in the same group cannot be activated simultaneously (i.e. $h_i = h_j = 1$) and hidden units in different groups have no dependency. Note that it is possible that no units is active, which is denoted as the ground state. By setting the between-hidden-neuron weights, the neurons with the highest activation value will be enlarged, while others squashed. As with the traditional Boltzmann machine, temperature may be added to intensify or temper competition. The special Boltzmann machine here is named competitive group-restricted Boltzmann machine.

4.2 GSMax Activation Function

Given that during training, what we are interested in is the posterior probability of hidden neurons being active, and we have purposefully kept the inference simple, the proposed prior simplified to an activation function, and can be put in any intermediate layers of neural network, and trained with back propagation, which we prove in this section. We denote this new activation function as Group Softmax (GSMax).

Denote \mathbf{h}_g as the vector consisting of hidden units in group g , and \mathbf{h}_{-g} as that of hidden units outside group g . According to the Boltzmann machine defined above, since $W_{ij}^h = 0$ when i, j are not in the same group, \mathbf{h}_g and \mathbf{h}_{-g} are independent given visible units \mathbf{v} . Therefore, we have

$$p(\mathbf{h}|\mathbf{v}) = p(\mathbf{h}_g|\mathbf{v})p(\mathbf{h}_{-g}|\mathbf{v}),$$

$$p(\mathbf{h}_g|\mathbf{v}) = \frac{1}{Z_g} e^{\mathbf{b}_g^T \mathbf{h}_g + \mathbf{v}^T W_g^v \mathbf{h}_g + \mathbf{h}_g^T W_g^h \mathbf{h}_g}.$$

\mathbf{h}_g is a binary vector representing the dimensions of \mathbf{h} that corresponds to units in group g . Its length is equal to the group size of g . \mathbf{b}_g is a vector composed of the elements of \mathbf{b} that correspond to units in group g . \mathbf{W}_g^v is a matrix composed of the columns \mathbf{W}^v that corresponds to units in group g . Similarly, \mathbf{W}_g^h is submatrix of \mathbf{W}^h and its rows and columns correspond to the units in group g .

Since $W_{ij}^h = -\infty$ when i, j are in the same group, $e^{\mathbf{h}_g^T W_g^h \mathbf{h}_g} = 0$ and therefore $p(\mathbf{h}_g|\mathbf{v}) = 0$, as long as \mathbf{h}_g has more than one elements equal to 1, i.e. more than one hidden units being activated.

\mathbf{h}_g has to be one of $\{\mathbf{e}_i\}_{i=0}^{|g|}$. The dimensionality of each \mathbf{e}_i is $|g|$. \mathbf{e}_0 is an all-zero vector. \mathbf{e}_i has the i th element equal to 1 and all others equal to 0. Denote $p(\mathbf{h}_{gi} = 1|\mathbf{v})$ as the posterior probability of hidden unit i in group g being active given visible units \mathbf{v} , and it's also the response on neuron i after GSMax. We have

$$p(\mathbf{h}_{gi} = 1|\mathbf{v}) = \frac{p(\mathbf{h}_g = \mathbf{e}_i|\mathbf{v})}{\sum_{k=0}^{|g|} p(\mathbf{h}_g = \mathbf{e}_k|\mathbf{v})} = \frac{e^{b_{gi} + \mathbf{v}^T W_{gi}^v}}{1 + \sum_{k=1}^{|g|} e^{b_{gk} + \mathbf{v}^T W_{gk}^v}}$$

where W_{gi}^v represents i th column of \mathbf{W}_g^v .

In conclusion, we can see the posterior probability distribution of \mathbf{h}_g over $\{\mathbf{e}_i\}$ given the input \mathbf{v} is a Softmax function augmented with a ground state that indicates no active hidden neurons.

5 Related works

In this section, we review in detail existing works that use unsupervised priors to associate complex patterns with internal neurons of neural networks. The network used is either trained in a supervised manner, or simplified from a supervised network for the purpose of investigation.

[12] puts forward a simplified convolutional network with clear mathematical interpretation in each of its components named Scattering Network (ScatNet). ScatNet hierarchically cascades handcrafted wavelet filters (defined on translation, rotation and scale groups), modulus non-linearity and subsampling averaging pooling, but does not combine channels. A hidden neuron (the activation after pooling) in ScatNet could be understood as a representation of a concept that is invariant to actions of the group, so it associates patterns of the input samples that are group symmetries of each other. While without pooling, each hidden neuron associates with a variation (a symmetry) of a certain prototypical sample. In [13], they demonstrate that if limited to two layers, features obtained by scattering transform are able to perform as well as a supervised trained neural network on object recognition datasets. [14] proves that an invariant and selective (with respect to a group) signature of an image could be obtained through pooling. Similarly, that signature associates a hidden neuron with patterns that are group symmetries of that sample. To investigate suitable mathematical constructions that associate patterns in the higher layers, [15] and [5] consider semidirect product of groups, while [16] considers reproducing kernels over probability distribution.

[17] [18] [19] apply pre-defined affine transformation groups on feature maps (or equivalent filters), to extend convolutional neural network beyond translation group and achieve good results.

Complementary to the above approaches, in this work, we work from endorsing priors on the relationship between elements of orbits to let filters associate with known variations of concepts, then analyze the learned neurons and filters.

6 Experiments

In this section, we train supervised neural network with the proposed association prior and investigate learned filters and neurons. First, we look at filters learned in the first layer, then at the neurons of the penultimate layer. Note that the two sections below uses different networks, the details of which are given in the appendices.

6.1 First layer discovers affine groups

In this section, we trained a commonly used supervised neural network on CIFAR10 [20] with the proposed prior applied to each layer, and visualize the learned filters of the first layer to explore learned filters. To make the competition among group elements sharply contrasted, we choose the first layer’s group size as 2. Please refer to the appendices for detailed network architecture and hyperparameters.

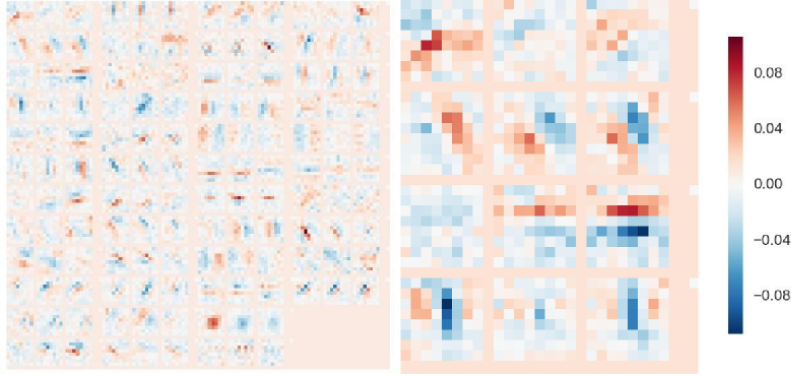


Figure 2: Left: visualize groups of filters learned in the first layer of the network trained on CIFAR10. The group size is 2. Each filter has three channels. Three horizontally consecutive patches in the same row are three 2D-filters of a 3D filter. Two vertically consecutive filters in the same column are in the same group. Right: Two groups are selected and plotted in a larger size to better visualize the details.

Figure 2 plots several groups of filters learned in the first layer of the network. It is observed that filters in the same group tend to capture orthogonal visual patterns: a vertical edge tends to be in the same group with a horizontal edge; a diagonal edge in the same group with an anti-diagonal edge and so on. Therefore, they are complementary. The relationship between those complementary filters possibly forms symmetry groups. For instance, the relationship between a vertical edge and a horizontal edge aforementioned may form a rotation group, thus a hidden neuron is able to associate with a rotation/variation of a filter. Though previously affine groups such as rotation group have also been observed in traditional neural network, but we cannot anticipate or control the relationship between which filters may form a symmetry group. While here the forming of groups conforms with the way we divide neurons into groups. It is interesting that by endorsing a statistics prior and purely through learning, neural networks can associate physical variations with filters. In this case, the cluster of variations of an edge is the lowest level concepts a neural network learns.

6.2 Penultimate neurons implicitly learns sub-class concepts

In this section, we investigate the filter learned in the higher layer. Due to the composition nature of neurons in the higher layer, it is hard to deduce or visualize learned filters corresponding to neurons. To work around, we look at concepts that are represented by neurons, and work on a problem where the invariant relationship between orbit elements are the membership of being in the same super-class. In the problem, samples have a two-layer hierarchical labels — a super-class label and a sub-class label. We apply the proposed prior and find the network is able to associate neurons with sub-class concepts when trained only with the super-class labels. While without the prior proposed, no clusters of sub-class concepts emerge.

6.2.1 Methods

In the problem, each super-class consists of sub-classes that do not overlap with sub-classes of other super-classes, and each sample does not belong to more than one sub-class or super-class. Dataset conforms with the above requirements are CIFAR100 [20]. We also build a similar dataset with CIFAR100 from ImageNet [21].

We apply GSMax on the penultimate layer, and compute representation of super-class concepts through the degenerated linear filter described in Section 3.2. The group division accords with the super-class structure. For example, in CIFAR100, the penultimate layer has 100 neurons, the last layer has 20 neurons, and neurons that belong to the same super class are in the same group. Max pooling are performed by groups to build representations of concepts of super-classes. The network is trained with only super-class labels. The network architecture we use is wide residual network [22]. To investigate the effect of the association prior endorsed by GSMax, we also did a control experiment. It has the same architecture, but with no GSMax association prior applied on the penultimate layer. To make the result easier to relate to current techniques, we also train a Alex-Net [2] style network. Please refer to the appendices for network architecture, hyperparameters and details of the new dataset.

After finishing training the network, to associate hidden neurons with sub-class concepts, we find the sub-class label that was mapped to a neuron most of the times in all forward propagation, adjusted by super-class labels. More specifically, each neuron in the penultimate layer has a list of sub-class labels mapped to it. Initially, the list was empty. A forward propagation was done for all test samples. For each sample, we got the activation of the penultimate layer, and extracted neurons that correspond to sub-classes of the super class label of this sample. Then we found the neuron that has the largest activation value, and put the sub-class label of this sample into the neuron’s list. After all sampled had been propagated, the sub-class label with the largest frequency in the list is taken as the true sub-class of this neuron.

6.2.2 Results

Architecture	CIFAR-100	Small ImageNet
Maxout	61.43	-
AlexNet	50.31 \pm 0.15	47.16 \pm 0.41
Implicit Learned	43.96 \pm 0.85	35.40 \pm 0.62
Control Group	22.47 \pm 3.26	9.83 \pm 1.76
Chance Level	20.00	17.33

Table 1: Accuracy of implicitly learned clusters of sub-classes (in bold), and comparison between implicitly learned clusters of sub-classes without GSMax (control group), chance level, currently supervised techniques, Alex Net and Maxout Network. Accuracy of all networks is reported with standard deviations for three repeated experiments, except that the result of Maxout network is directly retrieved from the paper. ‘-’ means the result is not available.

The results are summarized in Table 1. Considering that we mask out neurons that are not in the super-class of the propagated sample when mapping neurons to sub-classes, the chance level of classifying a sample to its right sub-class is calculated by averaging inverse of the sub-class number in each super class. As we could see, without any sub-class labels during training, the neurons in the penultimate layer get a far beyond chance level accuracy. It proves a network is able to associate a hidden neuron with a sub-class variation in a super-class concept with only association priors. Such result is further strengthened by the fact that without the GSMax association prior that characterizes relationship among the sub-class neurons, the accuracy is around chance level or drastically lower than chance level, implying no sub-class concepts emerges. The final accuracy even may be put at the end of the performance ladder of currently pure supervised approach, as we could see the accuracy gap between implicitly learned clusters and Alex Net is on par with the gap between Alex Net and Maxout network [23].

7 Conclusion

We propose with proper priors, neural networks can associate neurons in the intermediate layers with real-world concepts. It implies accompanying top level labels, neural networks can discover known concepts from data without labels, which may lead to truly unsupervised learning. We developed a preliminary prior that fulfills this purpose. We verified it experimentally by endorsing the proposed prior on neural networks, and found the prior helps neural networks learn physical

variations, i.e. variations related to symmetry groups, and semantic variations, i.e. sub-class variations in a super-class.

The proposed prior is still a crude characterization of relationship between orbit elements to promote order during learning. An interesting future work is to explore more general and accurate priors that may lead to adaptive creation of concepts, when available neurons are inadequate to encode concepts in samples.

References

- [1] Y. LeCun, Y. Bengio, *et al.*, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] A. Krizhevsky, I. Sutskever, *et al.*, “ImageNet Classification with Deep Convolutional Neural Networks,” in *NIPS*, 2012.
- [3] K. He, X. Zhang, *et al.*, “Identity Mappings in Deep Residual Networks,” in *ECCV*, 2016.
- [4] W. Miller, *Symmetry groups and their applications*. Academic Press, 1972.
- [5] S. Mallat, “Understanding deep convolutional networks.,” *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 374, no. 2065, 2016.
- [6] P. P. Brahma, D. Wu, *et al.*, “Why Deep Learning Works: A Manifold Disentanglement Perspective,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2015.
- [7] J. R. Gardner, M. J. Kusner, *et al.*, “Deep Manifold Traversal: Changing Labels with Convolutional Features,” pp. 1–11, 2015.
- [8] G. Montufar and R. Pascanu, “On the number of linear regions of deep neural networks,” *NIPS*, 2014.
- [9] M. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *ECCV*, 2014.
- [10] X. Glorot, A. Bordes, *et al.*, “Deep Sparse Rectifier Neural Networks,” *AISTATS*, 2011.
- [11] M. Carandini and D. J. Heeger, “Normalization as a canonical neural computation.,” *Nature reviews. Neuroscience*, vol. 13, no. 1, 2012.
- [12] S. Mallat, “Group Invariant Scattering,” *Communications on Pure and Applied Mathematics*, vol. LXV, pp. 1331–1398, jan 2012.
- [13] E. Oyallon, S. Mallat, *et al.*, “Generic Deep Networks with Wavelet Scattering,” in *ICLR Workshop*, 2014.
- [14] F. Anselmi, J. Z. Leibo, *et al.*, “Unsupervised learning of invariant representations,” *Theoretical Computer Science*, jun 2015.
- [15] J. Bruna, A. Szlam, *et al.*, “Learning Stable Group Invariant Representations with Convolutional Networks,” in *ICLR*, 2013.
- [16] F. Anselmi, L. Rosasco, *et al.*, “On Invariance and Selectivity in Representation Learning,” *Information and Inference, Special Issue: Deep Learning*, may 2016.
- [17] R. Gens and P. M. Domingos, “Deep Symmetry Networks,” *NIPS*, 2014.
- [18] T. S. Cohen and M. Welling, “Group Equivariant Convolutional Networks,” in *ICML*, 2016.
- [19] S. Dieleman, J. De Fauw, *et al.*, “Exploiting Cyclic Symmetry in Convolutional Neural Networks,” in *ICML*, 2016.
- [20] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” tech. rep., 2009.
- [21] J. D. J. Deng, W. D. W. Dong, *et al.*, “ImageNet: A large-scale hierarchical image database,” *CVPR*, 2009.
- [22] S. Zagoruyko and N. Komodakis, “Wide Residual Networks,” tech. rep., 2016.
- [23] I. J. Goodfellow, D. Warde-Farley, *et al.*, “Maxout Networks,” in *ICML*, 2013.
- [24] I. Goodfellow and D. Warde-Farley, “Pylearn2: a machine learning research library,” tech. rep., 2013.
- [25] M. Abadi, A. Agarwal, *et al.*, “TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems,” tech. rep., 2015.

8 Appendices

8.1 Experiment details for section 4.1

Network is trained on CIFAR10 dataset. The image data are global contrast normalized and ZCA whitened as in [23]. We use exactly the same architecture with Maxout from [24]: 8C192-4MP2-8C384-4MP2-8C384-2MP2-F2500-F10¹. where 8C192 means convolution layer with kernel size 8 and output channel number 192; 4MP2 means max pooling with kernel size 4 and stride 2; F10 means a fully connected layer with output channel number 10. Activation function is applied after pooling layer. Dropout is applied to all the layers with keep ratio 0.5, except for the visible unit(0.8). Weight decay is used instead of max norm. The base learning rate is set to 1, and decays with a rate of 0.1 every 25 epoch. Momentum optimizer with momentum 0.5 is used. GSMax group sizes are [2, 11, 8, 50], and temperature is set to 0.5.

8.2 Experiment details for section 4.2

We experimented on the CIFAR100 dataset [20]. It is a dataset of hierarchical labels that exactly conforms to the requirement discussed in Section 6.2.1. It has 100 classes containing 600 images each, 500 as training images and 100 as testing images. The 100 classes are grouped into 20 super-classes. As in [23], we preprocess the data using global contrast normalization and ZCA whitening. During training, normally used mild data augmentation is done, aka translations and horizontal flipping.

We also build a new dataset from ImageNet. The dataset is similar with CIFAR100, and conforms with our requirement. It has 84321 training images and 3300 test images. The training images are from the training set of ImageNet Large Scale Visual Recognition Challenge 2012 and testing images are from its validation set. The small ImageNet has 66 sub-classes, and 10 super-classes overall. During training, we used bounding boxes when available. Images are resized to 32×32 . The same data augmentation with CIFAR100 are used. The dataset will be publicly available.

The two datasets used the same network. The network architecture is based on wide residual network from [22], which consists of an initial convolution layer, followed by three stages of $2n$ convolution layers using k_i filters at stage i , followed by a global pooling layer, an inner product layer with GSMax as its activation function, a Maxout layer and a Softmax layer ($6n + 4$ in total). The first convolution in each stage $i > 1$ uses a stride of 2, so the feature map sizes are 32, 16, 8 for each three stages. We use WRN-28-10, where $n = 4$, and $k_i = 160, 320, 640$. The first convolution layer has 16 filters. The neurons in the last inner product layer are divided into groups according to the super-class structure. In the case of CIFAR100, neurons are divided into 20 groups, each of which has 5 neurons. The maxout layer has the same group size. Similarly for the Small ImageNet dataset. The training procedure was reproduced as close as possible from [22] in Tensorflow [25]. Super-class labels are applied to the last Softmax Layer.

¹Filter numbers for GSMax is slightly adjusted to make group sizes divide output channel numbers